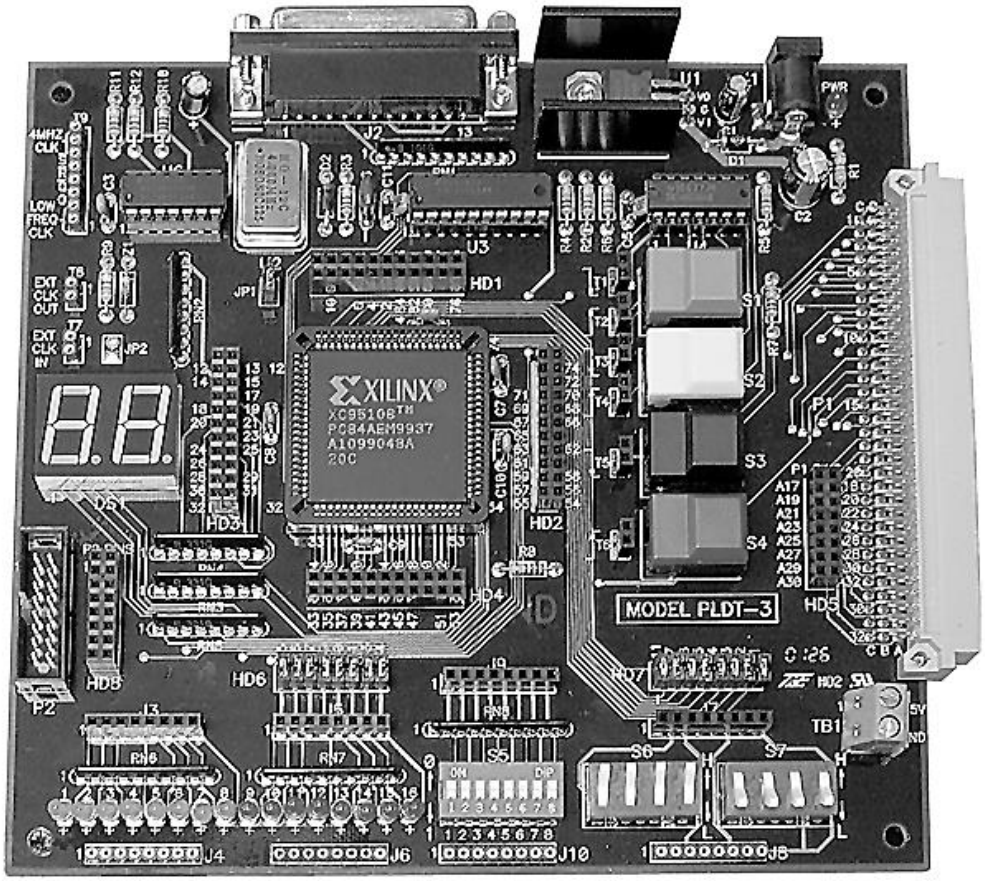
# PLDT-3 Trainer User Manual




© Copyright 2011
R.S.R. Electronics, Inc.
All rights reserved.

Ver. 1.0
04/11

# Table of Contents

**1.0 INTRODUCTION.....2**

**2.0 GENERAL DESCRIPTION.....4**

**3.0 BRIEF DESCRIPTION Of PLDT-3 BOARD.....5**

**4.0 DETAILED DESCRIPTION.....6**

**5.0 EXPERIMENTS.....11**

## 1.0 INTRODUCTION

The RSR® Electronics PLDT-3® digital logic trainer board, shown on cover page, has been designed as a "target board" for students and other users to design, implement, and test digital circuits using a modern programmable device and industry-standard design tools. It is built around the XILINX® Corporation XC95108 CPLD device in an 84-pin PLCC package.

While the fundamental principles of digital logic design have remained constant, the technology in which digital designs are implemented has changed rapidly. From transistors in the 1950s to small-scale integrated circuits (ICs) in the 1960s to large-scale ICs such as microprocessors in the 1970s, to the sophisticated programmable ICs of today

Early digital IC devices, such as 7400 series TTL, were initially built as standard building-blocks that had to be interconnected by copper traces on a circuit board. By the 1980s, the blocks could be mounted on one chip, and the interconnection done by "burning" a design into it. Those were the early programmable logic devices (PALs, PLAs, PLDs). They were one-time programmable (OTP), and required an electronic "box" called a device programmer to do the "burning". Later came devices that could be erased electrically and reprogrammed.

Today, ICs such as the venerable 7400 TTL chip are almost obsolete. Modern digital designs use large-scale programmable ICs such as field-programmable gate arrays (FPGAs) and complex programmable logic devices (CPLDs). Internally, FPGAs implement logic functions using look-up tables in memory blocks while CPLDs use sum-of-product terms configured from arrays of gates. Both FPGAs and CPLDs also contain flip-flops.

To program such devices, special computer-aided design (CAD) software is required that allows the user to enter a digital design on a PC, check it for validity, simulate its performance, and then download it into the target chip. Depending on the CAD package being used, a design can be entered using schematic capture software to draw a diagram showing interconnected gate symbols. Or a design may be entered using a text editor to create a file containing a set of Boolean equations written in a hardware description language (HDL). A common HDL is ABEL (Advanced Boolean Expression Language). Either way, the CAD software compiles the design into the low-level commands needed to configure the device.

The XC95108 CPLD contains 108 macrocells with 2400 usable gates arranged in six 36V18 function blocks as shown in ***Figure 1***. The chip is available in speeds from 7.5 ns to 20 ns delay time pin-to-pin. Outputs can drive 24 mA loads. The XC95108 is rated for a minimum of 10,000 program/erase cycles, and will hold a program for a minimum of 20 years.

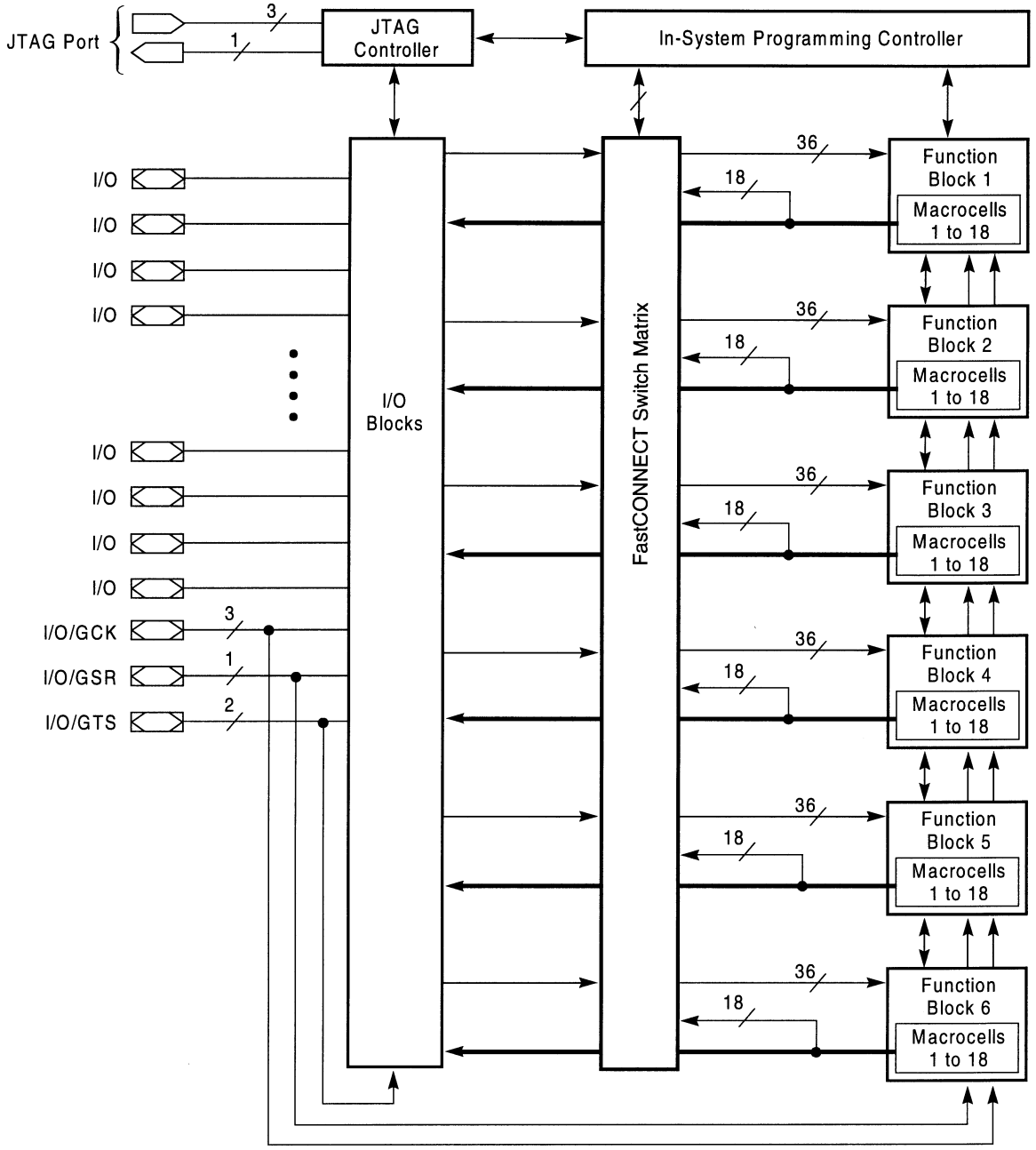

JTAG Port
3
1
JTAG Controller
In-System Programming Controller
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O
I/O/GCK
I/O/GSR
I/O/GTS
3
1
2
I/O Blocks
FastCONNECT Switch Matrix
36
18
Function Block 1
Macrocells 1 to 18
Function Block 2
Macrocells 1 to 18
Function Block 3
Macrocells 1 to 18
Function Block 4
Macrocells 1 to 18
Function Block 5
Macrocells 1 to 18
Function Block 6
Macrocells 1 to 18

*Figure 1*

## 2.0 GENERAL DESCRIPTION

The XC95108 used in the PLDT-3 digital logic trainer is in-service programmable (ISP). ISP means that the CPLD can be erased and reprogrammed while it is in a circuit, so a separate device-programmer box is not required. The PLDT-3 is powered by a wall-mounted power module which comes with the trainer.

The PLDT-3 connects to the parallel port of a personal computer via a standard 25-wire cable with DB25 connectors at both ends (***Figure 2**, detail 8).* The cable is part of the package when purchasing the PLDT-3. Standard design software, such as XILINX Foundation or Xilinx webpack, can be used.

Once commands are down-loaded from the PC to the CPLD, the cable may be disconnected; the CPLD "remembers" the design. Or the cable can be left attached for rapid reconfiguration of the logic design. Either way, the digital circuit programmed into the CPLD can be tested using the on-board switches and LEDs of the PLDT-3 board. Most CPLD pins are brought out via connector blocks. An uncommitted 16-pin male header is available for a user-defined ribbon-cable connection. Power, ground, and most CPLD pins are brought out to a 96-pin DIN-connector (P1), making the full power of the XC95108 available for advanced designs.

The 16 switches (8 toggle, 8 DIP) and 16 LEDs (8 red, 8 green) of the PLDT-3 can be accessed via jumper-blocks as well as via pin-jacks (***Figure 2**, details 1, 2, 3)* to allow connection of external circuits. Thus you can interface to external circuits built with small-scale ICs, such as logic chips, micro-controllers, memory chips, A/D & D/A converters, and so forth. Also on board the PLDT-3 are 4 momentary push-button switches (2 debounced and 2 not), an on-board 1 Hz clock, a 4 MHz clock module, a buffered input for an external clock, two 7-segment displays, and an SR flip-flop with inputs and outputs available via pin-jacks. A terminal-block allows access to +5 Volts and ground.

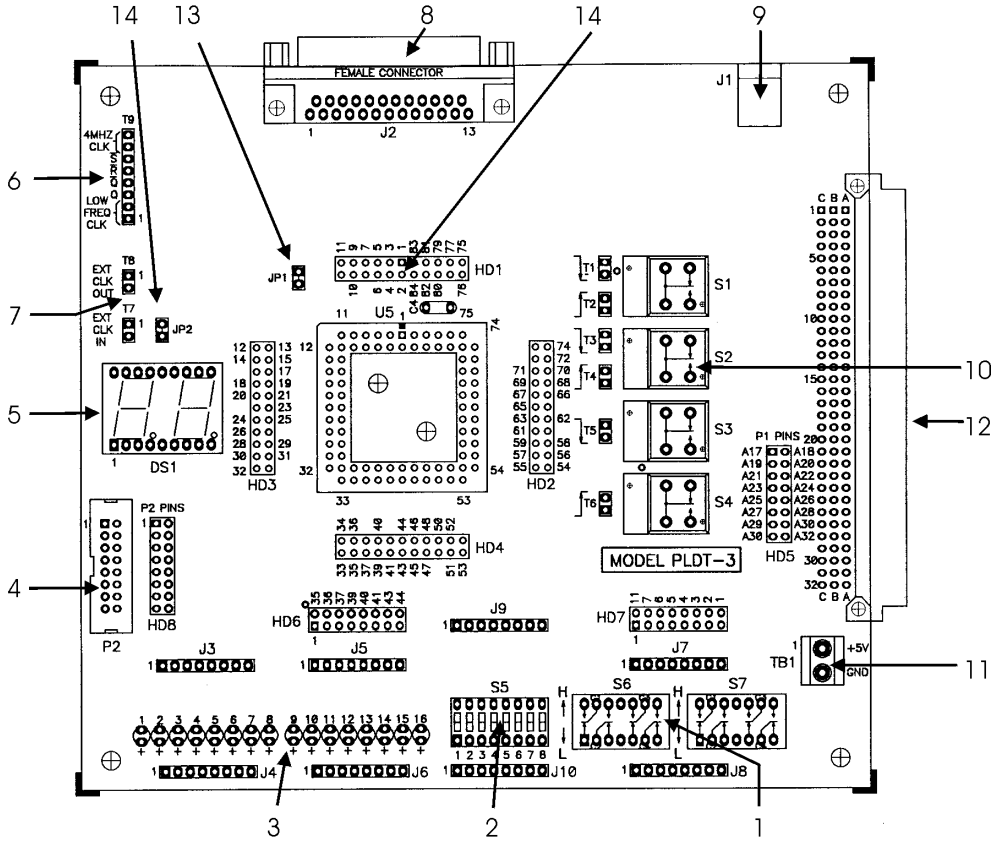



*Figure 2*

## 3.0 BRIEF DESCRIPTION OF PLDT-3 BOARD

### (*refer to Figure 3*)

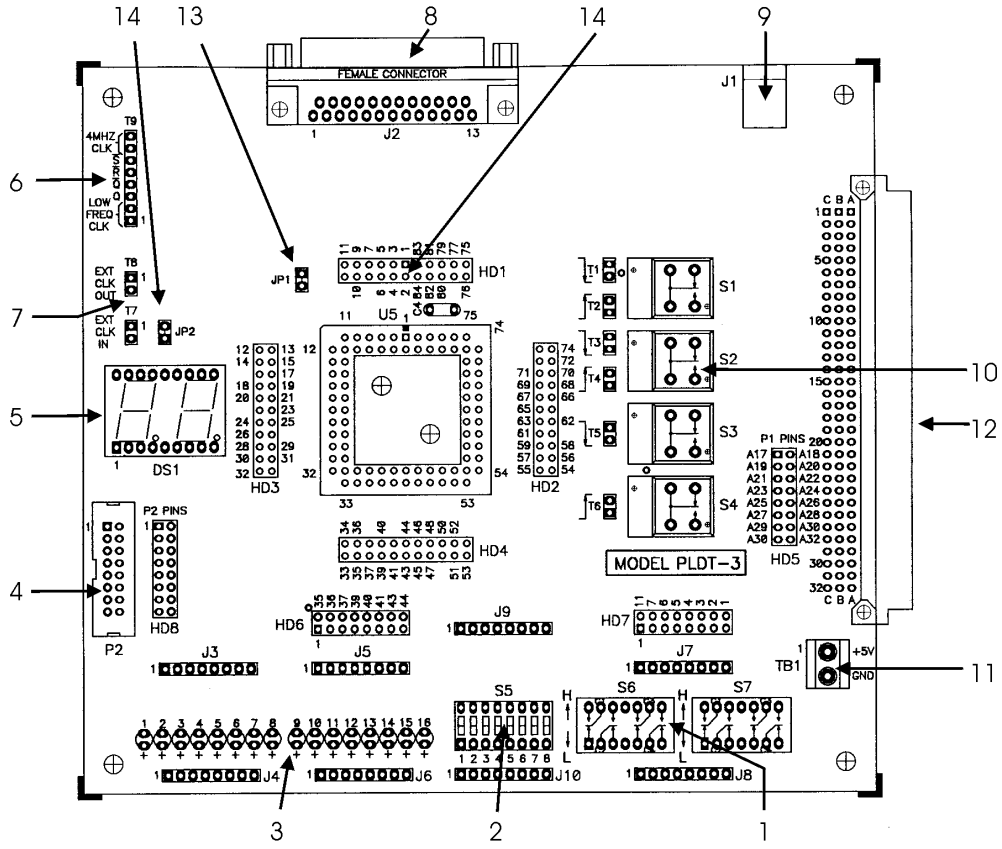



*Figure 3*

1) Toggle Switches **S6** and **S7** provide TTL-level signals to connectors **J7** and **J8.**
   The switches are also connected via jumper header **HD7** to specific CPLD pins (see table in **Sec. 4.0**). *NOTE: REMOVE THE JUMPERS FROM HD7 BEFORE DOWNLOADING TO CPLD.*
2) **J9** and **J10** are in parallel or connect to +5V and GND via 8-position DIP-switch **S5**. They are uncommitted signals: not hardwired to CPLD pins.
3) **J3** and **J4** are in parallel and connect to GREEN LEDs 1 – 8. TTL-level signals can drive the LEDs. The LEDs are uncommitted (not hardwired to CPLD pins).
   **J5** and **J6** are in parallel and connect to RED LEDs 9 – 16. TTL-level signals can drive the LEDs. The LEDs are hardwired to CPLD pins via jumper header **HD6**.
4) Shrouded 16-pin male socket **P2** pins are connected to corresponding pins of 16-pin female header **HD8**. The pins are uncommitted (not hardwired to CPLD pins).
5) Dual digit 7-seg display pins are hardwired to specific CPLD pins (see tables in **Sec. 4.0**). Display can be disconnected by cutting etch at **JP2**.
6) Connector **T9** provides access to on-board signals: Low frequency (1Hz) TTL clock, 4 MHz TTL clock, and $\overline{S}$, $\overline{R}$, Q, $\overline{Q}$ of SR flip-flop.
7) Clock conditioning circuit:
   **T7** will accept external sine-wave, square-wave, CMOS or TTL signal.
   **T8** provides TTL output of conditioned signal
8) Female DB-25 connector uses printer-cable to connect to parallel port.
9) Power jack: **J1**
10) Push buttons switches:
    **S1:** debounced, TTL-level output
    **T1:** falling edge **T2:** rising edge
    **S2:** debounced, TTL-level output
    **T3:** falling edge **T4:** rising edge
    **S3:** Normally-High TTL-level output at **T5**
    **S4:** Normally-Low TTL-level output at **T6**
11) **TB1** terminal block provides access to regulated +5 VDC and GND.
12) 96-pin DIN connector provides access to many CPLD pins, LEDs, Switches, +5 VDC and GND, and 16 uncommitted pins on header **HD5** (See table in **Sec. 4.0**)
13) **JP1** header jumper connects 4MHz TTL clock to pin 9 of CPLD chip.
14) Dual-row socket headers **HD1** through **HD4** allow access to all CPLD pins except power pins.

## 4.0 DETAILED DESCRIPTION

**(*Refer to Figure 2*)**

## 4.1 POWER

Power is supplied to the PLDT-3 from a wall mounted 9 Volt DC power supply connected to **J1** *(see detail 9)*. Note that the center pin of **J1** is positive. The 9 Volts is regulated down to 5 Volts by an on-board 7805 regulator. A terminal-block allows access to +5 Volts and ground (see detail 11). We recommend that you use the 9 Volt power-moduled supplied with the unit.

## 4.2 SWITCHES

Eight SPDT toggle switches, are provided by two DIP mounted switch modules, **S6** and **S7** *(see detail 1)*. When a toggle switch is in the position marked H (high), the switch pole is connected +5 Volts. In the position marked L (low), the switch pole is connected to ground. The switches are wired to pin-jacks on **J8** as well as to jumper-block **J7**. Jumpers on **HD7** connect the toggle switches to the CPLD as shown in the table below. Removing a jumper disconnects the corresponding switch from the CPLD.

For additional flexibility, an uncommitted 8-postion DIP switch (**S5**) is connected to pin-jacks on **J10** and to the jumper-block **J9**. The DIP switches are SPST with 1k pull-up resistors to the 5 Volt rail. When a DIP switch is slid up (into the ON position) its output goes to ground.

| **TOGGLE SWITCH CONNECTIONS** *(see detail **1** on **Figure 2**)* | | |
|---|---|---|
| **SWITCH** | **CPLD PIN** | **P1 PIN** |
| S6-1 | 11 | C25 |
| S6-2 | 7 | C26 |
| S6-3 | 6 | C27 |
| S6-4 | 5 | C28 |
| S7-1 | 4 | C29 |
| S7-2 | 3 | C30 |
| S7-3 | 2 | C31 |
| S7-4 | 1 | C29 |

**CAUTION:** When "**HD7**" jumpers (shorting bars) are connected, eight SPDT toggle switches in switch modules **S6** and **S7** will provide either **GND** ("0") or **+5V** ("1") to the corresponding pins of CPLD. This is shown in ***Figure 4***. This can cause permanent damage to CPLD unless the corresponding pins of CPLD are programmed as inputs in your design. So please make sure to remove all eight jumpers from "**HD7**" before programming your design into CPLD chip. When using "**S6**" and **S7**" as inputs in your design, you have to program the corresponding CPLD pins as input. These pins are 1, 2, 3, 4, 5, 6, 7, and 11. Only then connect jumpers for corresponding pins on header "**HD7**"

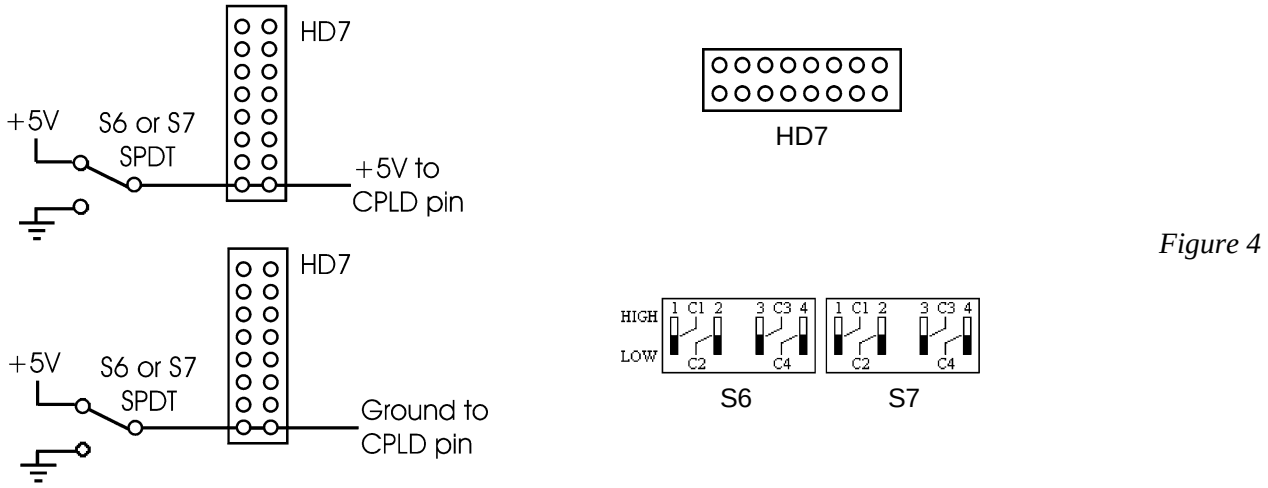



*Figure 4*

## 4.3 MOMENTARY SWITCHES

The two debounced momentary push-button switch, **S1** and **S2** (*detail 10*), can be used to apply manually generated input pulses to pins of the CPLD. For **S1**, the falling-edge is available at pin jack **T1** while the rising edge is available at **T2**. For **S2**, the falling-edge is available at pin jack **T3** while the rising edge is available at **T4**.

The two non-debounced momentary push-button switch, **S3** and **S4** (*detail 10*), also can be used to apply manually generated input pulses to pins of the CPLD. **S3** generates a falling edge available at pin jack **T5** while **S4** generates a rising edge available at **T6**. Note that CPLD pin 9 is a global clock.

## 4.4 CLOCK SIGNALS

### 4.4.1 External Clock Input

TTL clock signal from a function generator can be brought onto the board via jack labeled "EXT CLK IN" on the **T7** pin-jacks *(detail 7)*. CLK IN is buffered by a 74HC00 and the buffered clock is available on the **T8** pin-jacks.

### 4.4.2 The Low-Frequency Clock

An on-board clock of approximately 1 Hertz is available at pin-jacks on **T9** *(see detail 6)*. It is useful when, for example, you wish to watch LEDs driven by an up-counter.

### 4.4.3 The 4 MHz Clock Module

A 4 MHz clock oscillator module is connected to pin 9 of the CPLD through the jumper at **JP1**. Removing the jumper disconnects the module from the CPLD, but the module continues to oscillate. The 4 MHz clock is available at **T9** *(see detail 6)*.

## 4.5 LEDs

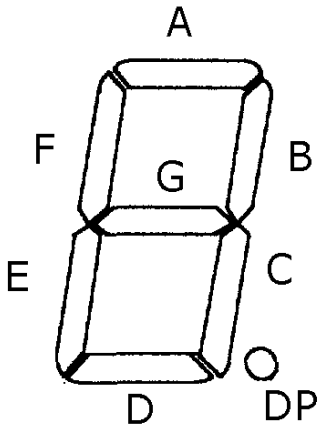
*See detail 3*. Eight green LEDs numbered 1 to 8 are connected to pins-jacks on **J4** and to jumper-block **J3**. Eight red leds numbered 9 to 16 are connected to pins-jacks on **J6** and to jumper-block **J5**. The red LEDs are connected to pins on the CPLD through jumpers on **HD6** as shown in the table below. Removing a jumper disconnects the corresponding LED from the CPLD. With all jumpers removed from **HD6**, a ribbon cable could be connected to it to allow external access to the CPLD pins. LED current is limited by 330 Ohm resistors in **RN6** and **RN7**.

| **RED LED #** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** |
|---|---|---|---|---|---|---|---|---|
| **CPLD PIN** | **35** | **36** | **37** | **39** | **40** | **41** | **43** | **44** |

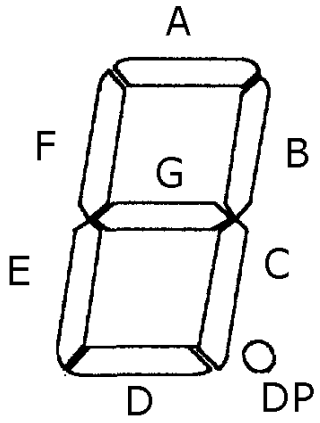
## 4.6 THE 7-SEGMENT DISPLAYS

Two common-cathode 7-segment displays *(detail 5)* are connected to the CPLD pins as shown in the table below. Current is limited by 330 Ohm resistors in **RN2**, **RN3**, **RN4**, and **RN5**.

| **7- SEGMENT DISPLAY** | | | |
|---|---|---|---|
| **TENS DIGIT SEGMENT** | **CPLD PIN** | **UNITS DIGIT SEGMENT** | **CPLD PIN** |
| A | 57 | A | 15 |
| B | 58 | B | 18 |
| C | 61 | C | 23 |
| D | 62 | D | 21 |
| E | 63 | E | 19 |
| F | 65 | F | 14 |
| G | 66 | G | 17 |
| DP | 67 | DP | 24 |




***Ten Digit Segment***



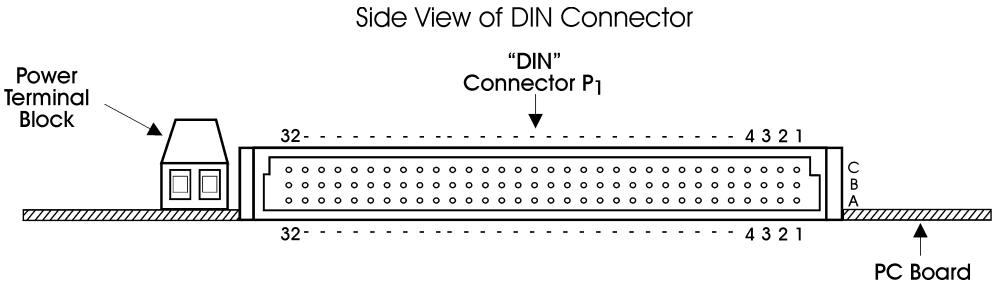

***Unit Digit Segment***

## 4.7 ON-BOARD OSCILLATOR

Resonator **X1** together with integrated circuit U1 form a 4 MHz oscillator driving pins 9 of the CPLD. The oscillator can be used for designs requiring a high-speed clock.

## 4.8 CONNECTORS:

Most of the CPLD pins are brought out to connector **P1**, **HD1**, **HD2**, **HD3**, **HD4** and **HD5**.




## "P1" Connector

| *Row A Pin #* | *Connected to:* |
|---|---|
| 1 | +5Vdc |
| 2 | GND |
| 3 | Pin # 12 of CPLD |
| 4 | Pin # 10 of CPLD |
| 5 | Pin # 84 of CPLD |
| 6 | Pin # 82 of CPLD |
| 7 | Pin # 80 of CPLD |
| 8 | Pin # 77 of CPLD |
| 9 | Pin # 75 of CPLD |
| 10 | Pin # 72 of CPLD |
| 11 | Pin # 69 of CPLD |
| 12 | Pin # 26 of CPLD |
| 13 | Pin # 31 of CPLD |
| 14 | Pin # 32 of CPLD |
| 15 | Pin # 33 of CPLD |
| 16 | Pin # 34 of CPLD |
| 17 | HD5 |
| 18 | HD5 |
| 19 | HD5 |
| 20 | HD5 |
| 21 | HD5 |
| 22 | HD5 |
| 23 | HD5 |
| 24 | HD5 |
| 25 | HD5 |
| 26 | HD5 |
| 27 | HD5 |
| 28 | HD5 |
| 29 | HD5 |
| 30 | HD5 |
| 31 | HD5 |
| 32 | HD5 |

| *Row B PIN #* | *Connected to:* |
|---|---|
| 1 | +5Vdc |
| 2 | GND |
| 3 | CLK IN of Header "T7" |
| 4 | Pin # 9 of CPLD |
| 5 | Pin # 83 of CPLD |
| 6 | Pin # 81 of CPLD |
| 7 | Pin # 79 of CPLD |
| 8 | Pin # 76 of CPLD |
| 9 | Pin # 74 of CPLD |
| 10 | Pin # 71 of CPLD |
| 11 | Pin # 25 of CPLD |
| 12 | Pin # 55 of CPLD |
| 13 | Pin # 53 of CPLD |
| 14 | Pin # 51 of CPLD |
| 15 | Pin # 47 of CPLD |
| 16 | Pin # 45 of CPLD |
| 17 | LED 16/ optional pin 44 of CPLD |
| 18 | LED 15/ optional pin 43 of CPLD |
| 19 | LED 14/ optional pin 41 of CPLD |
| 20 | LED 13/ optional pin 40 of CPLD |
| 21 | LED 12/ optional pin 39 of CPLD |
| 22 | LED 11/ optional pin 37 of CPLD |
| 23 | LED 10/ optional pin 36 of CPLD |
| 24 | LED 09/ optional pin 35 of CPLD |
| 25 | SW 1 of "S6"/ optional pin 11 of CPLD |
| 26 | SW 2 of "S6"/ optional pin 07 of CPLD |
| 27 | SW 3 of "S6"/ optional pin 06 of CPLD |
| 28 | SW 4 of "S6"/ optional pin 05 of CPLD |
| 29 | SW 1 of "S7"/ optional pin 04 of CPLD |
| 30 | SW 2 of "S7"/ optional pin 03 of CPLD |
| 31 | SW 3 of "S7"/ optional pin 02 of CPLD |
| 32 | SW 4 of "S7"/ optional pin 01 of CPLD |

| *Row C Pin #* | *Connected to:* |
|---|---|
| 1 | +5Vdc |
| 2 | GND |
| 3 | T1 Header Mom. Switch S1 |
| 4 | T2 Header Mom. Switch S1 |
| 5 | T3 Header Mom. Switch S2 |
| 6 | T4 Header Mom. Switch S2 |
| 7 | T5 Header Mom. Switch S3 |
| 8 | T6 Header Mom. Switch S4 |
| 9 | Pin # 70 of CPLD |
| 10 | Pin # 68 of CPLD |
| 11 | Pin # 56 of CPLD |
| 12 | Pin # 54 of CPLD |
| 13 | Pin # 52 of CPLD |
| 14 | Pin # 50 of CPLD |
| 15 | Pin # 48 of CPLD |
| 16 | Pin # 46 of CPLD |
| 17 | Pin # 44 of CPLD |
| 18 | Pin # 43 of CPLD |
| 19 | Pin # 41 of CPLD |
| 20 | Pin # 40 of CPLD |
| 21 | Pin # 39 of CPLD |
| 22 | Pin # 37 of CPLD |
| 23 | Pin # 36 of CPLD |
| 24 | Pin # 35 of CPLD |
| 25 | Pin # 11 of CPLD |
| 26 | Pin # 07 of CPLD |
| 27 | Pin # 06 of CPLD |
| 28 | Pin # 05 of CPLD |
| 29 | Pin # 04 of CPLD |
| 30 | Pin # 03 of CPLD |
| 31 | Pin # 02 of CPLD |
| 32 | Pin # 01 of CPLD |

### 4.8.1 Connectors HD1, HD2, HD3, HD4

Dual-row socket headers **HD1** through **HD4** allow easy access to all CPLD pins except power and **GND** pins

### 4.8.2 Connected HD5

Selected Pins of "**P1**" connector are brought out to header HD5.

### 4.8.3 Connector HD8 and P2

Shrouded 16-pin male socket **P2** pins are connected to corresponding pins of 16-pin female header **HD8**. The pins are uncommitted (not hardwired to CPLD pins).

## 5.0 EXPERIMENTS

***See appendix A for downloading and installing WebPack***

## 5.1 TTL EXPERIMENT: COMBINATIONAL LOGIC

A simple TTL experiment is shown below.

### EXPERIMENT: GATE CIRCUIT USING 74LS00 TTL CHIP

**OBJECTIVE**

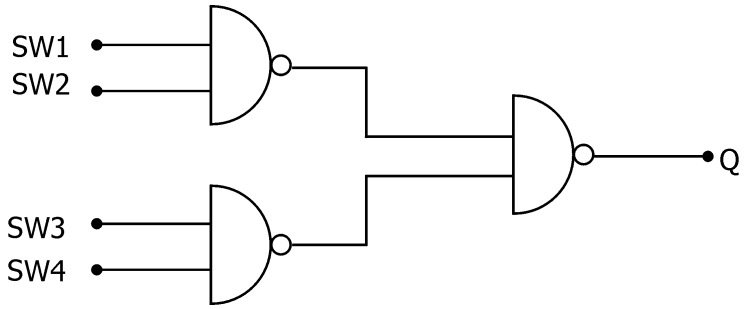
To examine a simple combinatorial circuit built with a TTL chip. S5 switches of the PLDT-3 will be used to generate input while the LED's 1-8 of the PLDT-3 will be used to show output.

**PARTS**

1. Solderless Breadboard
2. 74LS00 TTL Quad-2 Integrated Circuit
3. PLDT-3 with power module

**PROCEDURE**

A) Use Boolean Algebra to get an equation for Q from the following circuit:




Q = _____

B) Using the equation, fill in the truth table on the next page (just fill in the column labeled **Q**).

C) Build the circuit.

1. Insert the 74LS00 into the solderless breadboard.
2. Connect a wire from pin 7 of the 74LS00 to the terminal "**TB1**" labeled "**GND**" on the PLDT-3.
3. Connect a wire from pin 14 of the 74LS00 to the terminal "**TB1**" labeled "+5V" on the PLDT-3.
4. Connect a wire from pin 1 of the 74LS00 to the pin 1 of connector "**J10**" of the switch "**S5**".
5. Connect a wire from pin 2 of the 74LS00 to the pin 2 of connector "**J10**" of the switch "**S5**".
6. Connect a wire from pin 3 to pin 4 of the 74LS00.
7. Connect a wire from pin 13 of the 74LS00 to the pin 3 of the connector "**J10**" of the switch "**S5**".
8. Connect a wire from pin 12 of the 74LS00 to the pin 4 of the connector "**J10**" of the switch "**S5**".
9. Connect a wire from pin 11 to pin 5 of the 74LS00.
10. Connect a wire from pin 6 of the 74LS00 to the pin 1 of the connector "**J4**" of the LED1.

Connect the power module to the PLDT-3 and plug the power module into a 110 VAC outlet.

Assuming a switch at “H” is a logic “1” and a lit LED is a logic “1”, fill in the column labeled **LED** on the following truth-table.

| **SW 1** | **SW 2** | **SW 3** | **SW 4** | **Q from equation** | **LED On or Off** |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | | |
| 0 | 0 | 0 | 1 | | |
| 0 | 0 | 1 | 0 | | |
| 0 | 0 | 1 | 1 | | |
| 0 | 1 | 0 | 0 | | |
| 0 | 1 | 0 | 1 | | |
| 0 | 1 | 1 | 0 | | |
| 0 | 1 | 1 | 1 | | |
| 1 | 0 | 0 | 0 | | |
| 1 | 0 | 0 | 1 | | |
| 1 | 0 | 1 | 0 | | |
| 1 | 0 | 1 | 1 | | |
| 1 | 1 | 0 | 0 | | |
| 1 | 1 | 0 | 1 | | |
| 1 | 1 | 1 | 0 | | |
| 1 | 1 | 1 | 1 | | |

## 5.2 CPLD EXPERIMENT: BASIC GATES

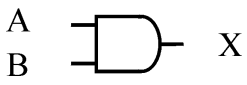
The objective of this experiment is to gain familiarity with the use of Xilinx Foundation Series software by designing and implementing a simple gate circuit.

AND gates and OR gates are two fundamental building blocks of digital circuits. Both AND and OR gates can have two or more inputs, but only one output. The input / output behavior of a gate is shown by a truth table and is written as a Boolean equation as shown below.

**Gate Characteristics:**

**The AND Gate**

**Symbol** **Boolean Equation** **Truth Table**

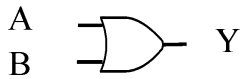



**X = AB**

| A | B | | X |
|---|---|---|---|
| 0 | 0 | | 0 |
| 0 | 1 | | 0 |
| 1 | 0 | | 0 |
| 1 | 1 | | 1 |

**The OR Gate**

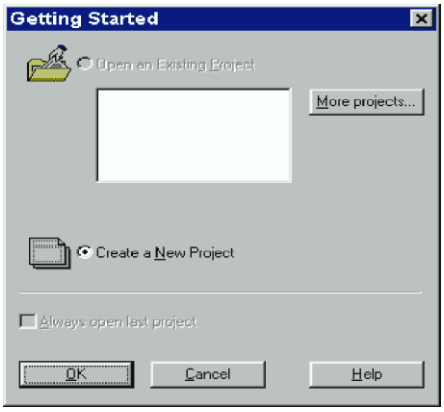
**Symbol** **Boolean Equation** **Truth Table**




**Y = A + B**

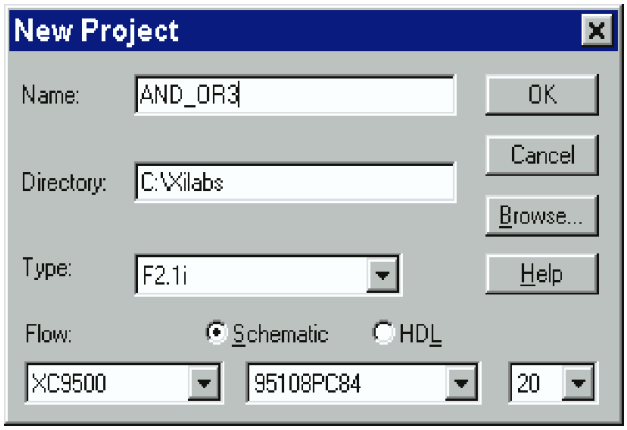
| A | B | | Y |
|---|---|---|---|
| 0 | 0 | | 0 |
| 0 | 1 | | 1 |
| 1 | 0 | | 1 |
| 1 | 1 | | 1 |

**PROCEDURE:**

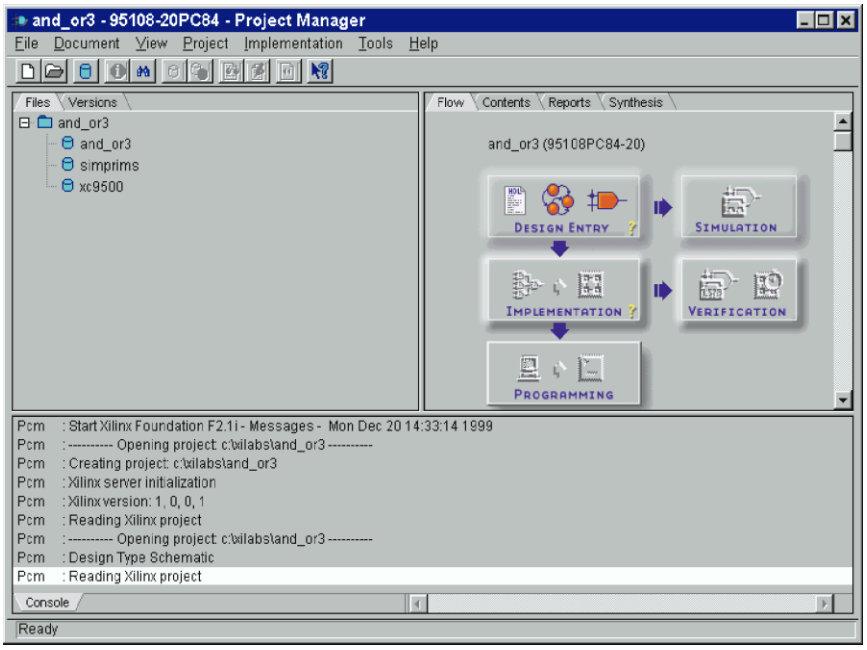
1. Create a new directory on your C: drive (call it XILABS, or you name it). You will save all your experiment files in this directory since they will be too big for a floppy.
2. Start the Software by clicking on the following: **Start → Programs → Xilinx Foundation Series → Xilinx Foundation Project Manager**, or just click on the Project Manager Icon on the Windows desk top.
3. 


In the **Getting Started** window choose **Create a New Project** and click on the **OK** button.

4. When the **New Project** window appears, type **AND_OR3** in the **Name** editor box as your project name. In the **Directory** edit box, browse to get **Xilabs**. Choose **Type** as **F2.1i**. Click on **Schematic**. Select **Family**, **Part**, and **Speed** to be **XC9500**, **95108PC84** and **20**, respectively. Then, click on **OK**.

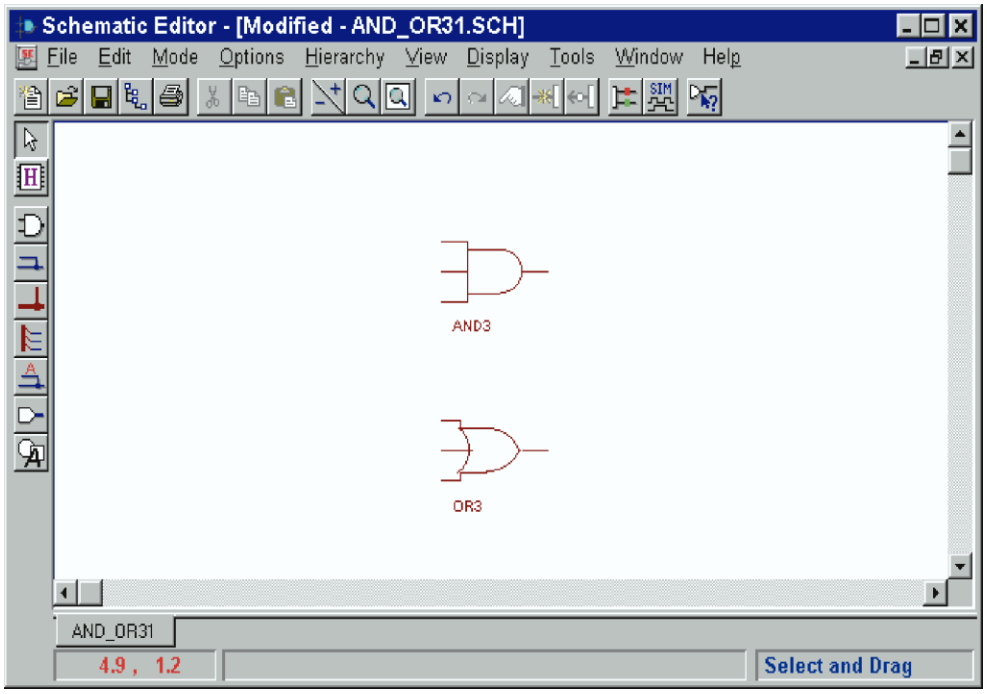


The new project is now created and you should see a window like this:



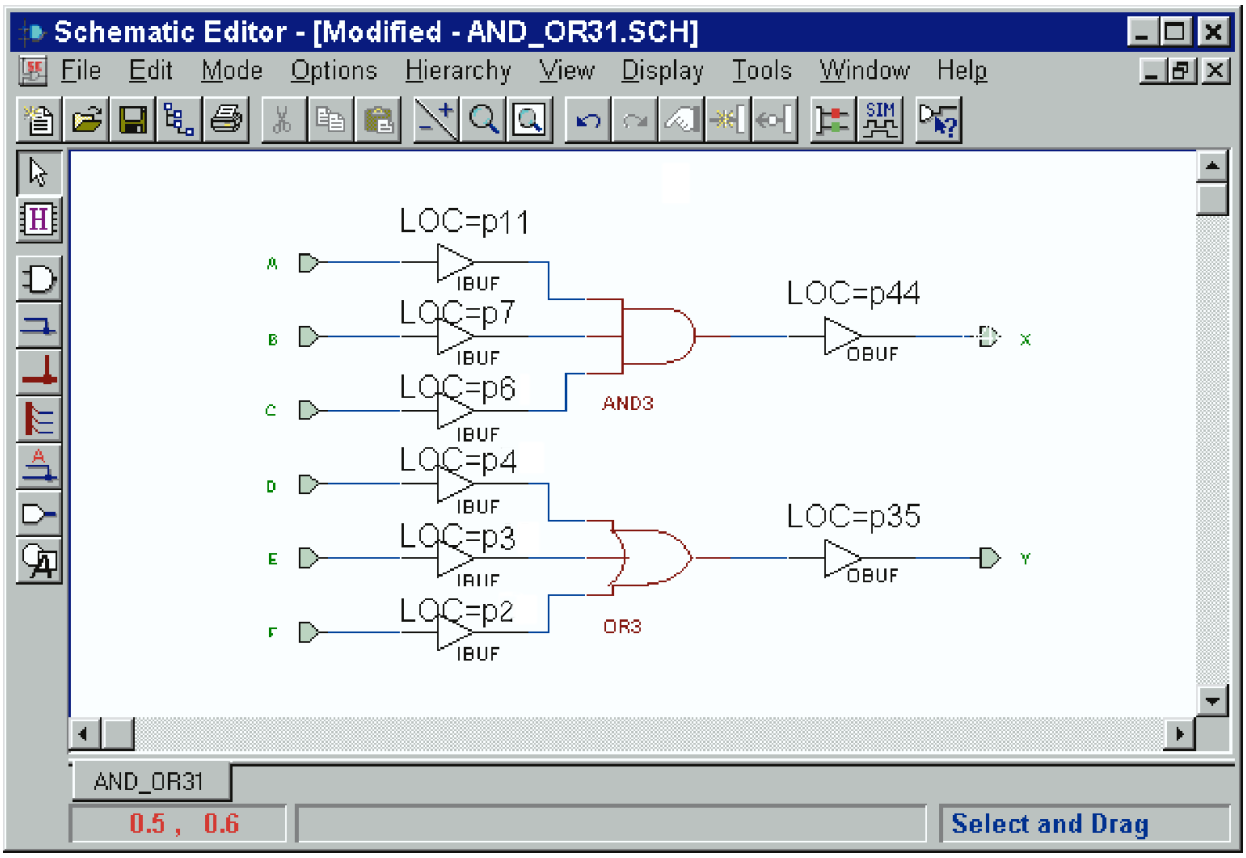

5. Click on the **Schematic Editor** button in the right pane to get the **Schematic Editor** window.

6 Click on the AND gate symbol to get the **SC Symbols** window. Then move the bar down to **AND3** in that window. Click on AND3. Go to the Schematic Editor window and click at where you want to drop the symbol. Repeat the process for the **OR3** gate symbol. The Schematic Editor should appear as below.




7. Save the schematic by clicking on **File → Save As.** In the Save As sub-window, edit the **File name** box to **AND_OR31.SCH** and click on the **OK** button to save the file to the AND_OR3 subdirectory. Add the IBUFs, OBUFs, and I/O terminals. Then wire all the connections.

8. Assign the six inputs A, B, C, and D, E, F to pins 11, 7, 6, and 4, 3, 2, respectively. That lets us use toggle switches 1, 2, 3, 5, 6 and 7 on the PLDT-3 board. Assign LED 10 and LED 9 to display the outputs X and Y ( X = pin 44 and Y = pin 35). To assign pin 11 to input A, double click the buffer for signal A. In the **Symbol Properties** sub-window, type **loc** in the **Name** editor box and **p6** in the **Description** editor box. Then, click on the **Add** button followed by the **Move** button. Finish all the pin assignments in the same way. Your Schematic Editor window should now look as shown below. Save your schematic by choosing **File → Save**.




9. Choose **Option → Create Netlist** from the menu bar. The message should say "Netlist created successfully". Click on the **OK** button.

10. Choose **Option → Integrity Test**. The message should say "Integrity test passed successfully". Click on the **OK** button to go back to the Schematic Editor.

11. Select **Option → Export Netlist**. For **Files of type** choose **Edif 200 [*.EDN].** Click on the **Open** button to export the file.

12. The AND_OR3 design in the Schematic Editor is now complete. Return to the Project Manager window by selecting **File →Exit**. Verify that a green check mark is on the Design Entry button.

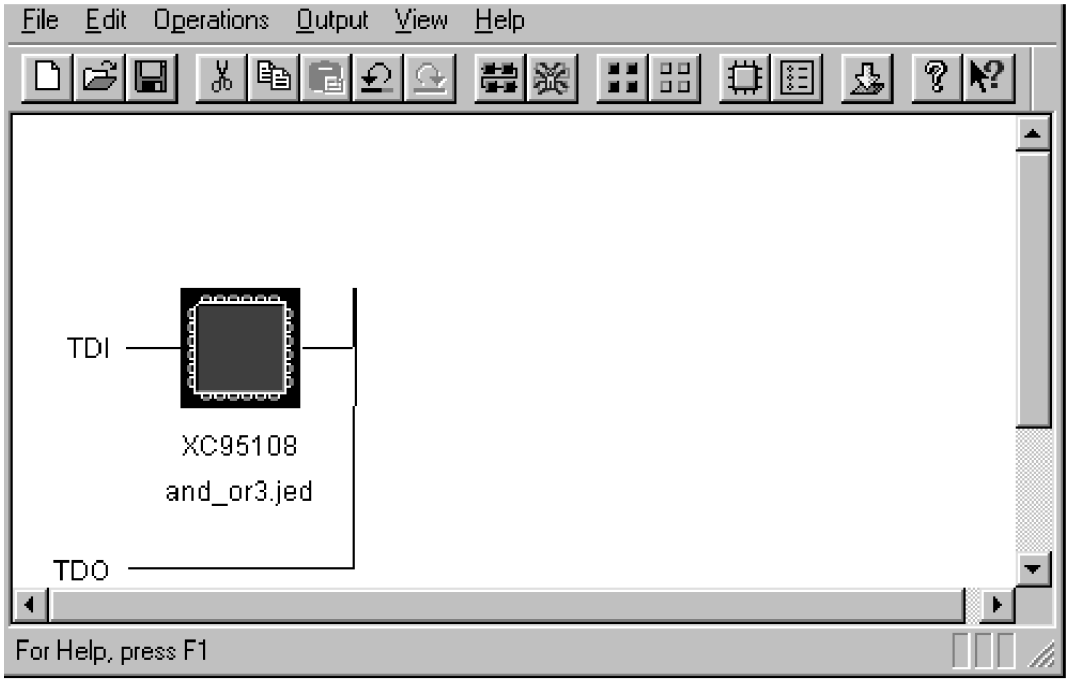
13. Click on the **Implementation** button to compile the design. Verify that the **Device**, **Speed**, **New version name** and **New revision name** edit boxes have ***95108PC84***, ***20***, ***ver1*** and ***rev1***, respectively. Click on the **Run** button.

14. If errors are detected during this stage, the process will be terminated with the message "Implementation completed with errors". Go back to the Schematic Editor and fix them. If there is no error, you will see a window as shown below. Use **File → Close** to go back to the Project Manager and verify that a green check mark appears on the Implementation button.

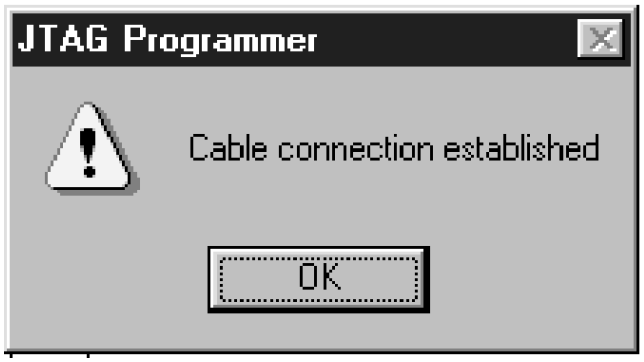
15. Click on **OK** in the message sub-window.

16. Plug the AC adapter into an outlet, and connect it to the PLDT-3 board. Then connect your PC to the PLDT-1 board with the parallel cable provided.

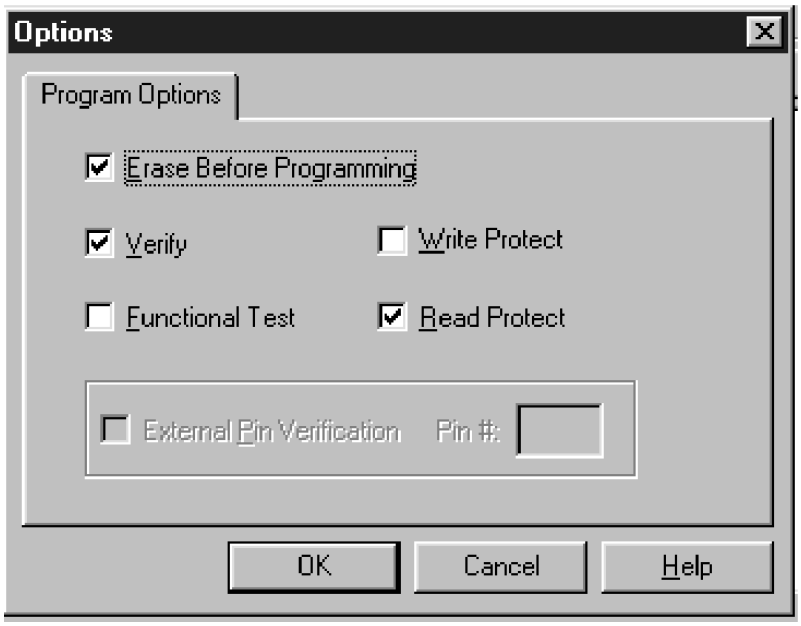
17. To download the program, click on the **Device Programming** button in the Project Manager window, or choose **Tools → Device Programming** on the menu at the top.




18 In the **JTAG Programmer** window, choose **Output →Cable Auto Connect.** You should see the window:



19 In the **JTAG Programmer** window, choose **Operations → Program** from the menu bar, the **Options** sub-window should appear. Select **Erase Before Programming** and **Verify**. Then click on the **OK** button.




20. When the download is finished, click on the **OK** button in the **Operation Status** sub-window to go back to the JTAG Programmer window. Then, choose **File → Exit**. At the message sub-window prompt, select **Yes**. In the **Save As** sub-window, click on **Yes** to save the JEDEC file and return to the Project Manager window.

21. Run the program on your PLDT-3 board and fill in the following truth-tables by using the switches for inputs and the LEDs for output. Remember to insert appropriate jumpers in connector block "**HD7**".

**Truth-Table for 3-Input AND Gate**

| A | B | C | | X = A • B • C |
|---|---|---|---|---|
| 0 | 0 | 0 | | |
| 0 | 0 | 1 | | |
| 0 | 1 | 0 | | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |

**Truth-Table for 3-Input OR Gate**

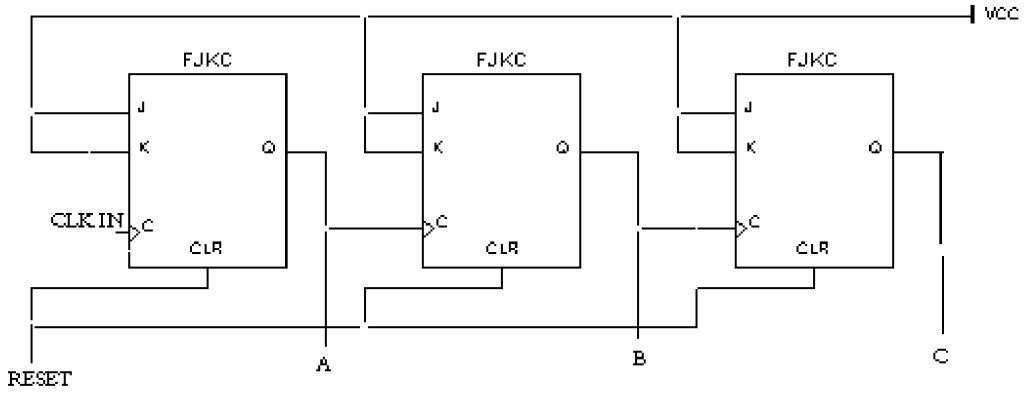
| D | E | F | | Y = D + E + F |
|---|---|---|---|---|
| 0 | 0 | 0 | | |
| 0 | 0 | 1 | | |
| 0 | 1 | 0 | | |
| 0 | 1 | 1 | | |
| 1 | 0 | 0 | | |
| 1 | 0 | 1 | | |
| 1 | 1 | 0 | | |
| 1 | 1 | 1 | | |

22. Do not forget to remove all jumpers (shorting bars) from connector block "**HD7**". In fact, it is a good practice to do this at the end of every experiment.

## 5.3 CPLD EXPERIMENT: ASYNCHRONOUS (RIPPLE) COUNTER

The objective of this experiment is to gain familiarity with the use of Xilinx Foundation Series software by designing and implementing an asynchronous counter using JK flip-flops (FFs).

In an asynchronous counter, the output of each FF serves as the CLK input signal for the next FF. The name asynchronous comes from the fact that all the FFs do not change states at the same time. After each clock, change "ripples down" the chain of FFs, which is why they also are called ripple counters.

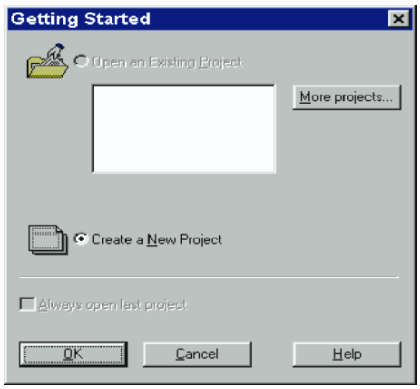


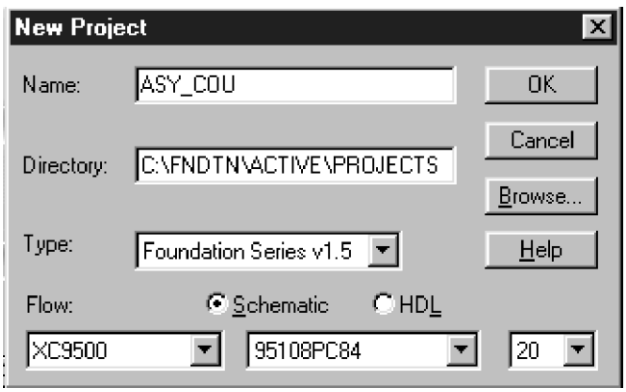

The clock pulses applied to CLKIN are generated by the PULSE switch.

| C | B | A | NUMBER OF PULSES |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 2 |
| 0 | 1 | 1 | 3 |
| 1 | 0 | 0 | 4 |
| 1 | 0 | 1 | 5 |
| 1 | 1 | 0 | 6 |
| 1 | 1 | 1 | 7 |

**PROCEDURE:**

1. Create a new directory on your C: drive (call it XILABS, or you name it). You will save all your experiment files in this directory since they will be too big for a floppy.
2. Start the Software by clicking on the following: **Start → Programs → Xilinx Foundation Series → Xilinx Foundation Project Manager**, or just click on the Project Manager Icon on the Windows desk top.
3. In the **Getting Started** window choose **Create a New Project** and click on the **OK** button.



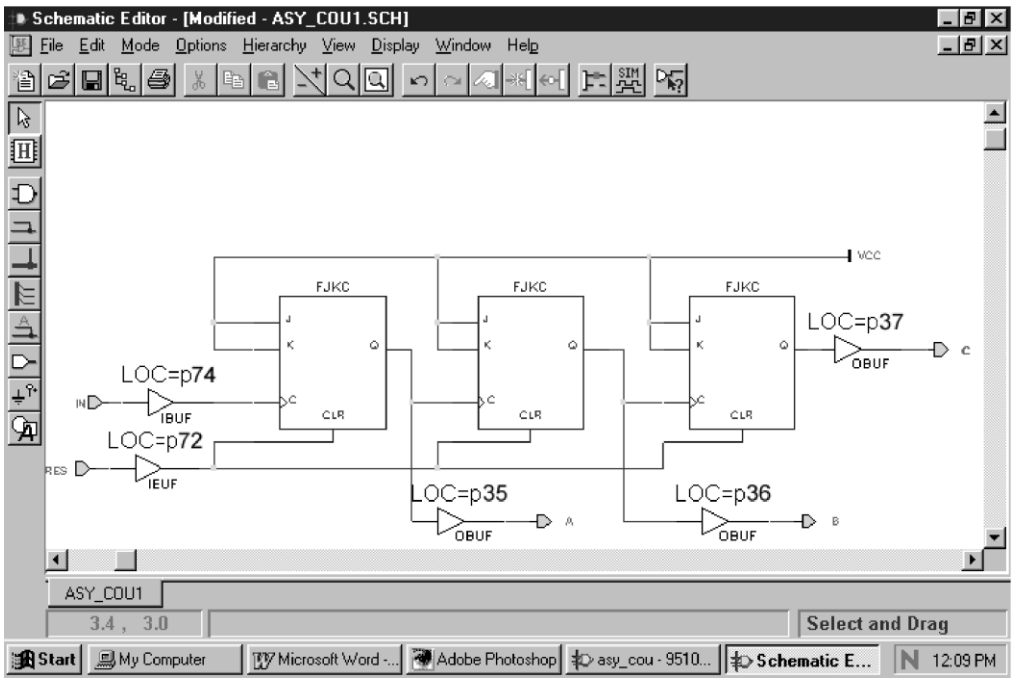
4. When the **New Project** window appears, type **ASY_COU** in the **Name** editor box as your project name. In the **Directory** edit box, browse to get **Xilabs**. Choose **Type** as **F2.1i**. Click on **Schematic**. Select **Family**, **Part**, and **Speed** to be **XC9500**, **95108PC84** and **20**, respectively. Then, click on **OK**. The new project is now created .

5. Click on the **Schematic Editor** button in the right pane to get the **Schematic Editor** window

6. Save the schematic by clicking on **File → Save As.** In the Save As sub-window, edit the **File name** box to **AND_OR31.SCH** and click on the **OK** button to save the file to the **ASY_COU** subdirectory. Add the IBUFs, OBUFs, and I/O terminals. Then wire all the connections.

7. As we did in the previous experiment:

   Assign the inputs IN and RES to pins 74 and 72 respectively . Assign LED 9 , LED 10 and LED 11 to display the outputs A, B, and C ( A = pin 35, B = pin 36, and C = pin 37). To assign pin 74 to IN, double click the buffer for signal IN. In the **Symbol_Properties** sub-window, type **loc** in the **Name** editor box and **p74** in the **Description** editor box. Then, click on the **Add** button followed by the **Move** button. Finish all the pin assignments in the same way. Your Schematic Editor window should now look as shown below. Save your schematic by choosing **File → Save**.




8. Choose **Option → Create Netlist** from the menu bar. The message should say "Netlist created successfully". Click on the **OK** button.

9. Choose **Option → Integrity Test**. The message should say "Integrity test passed successfully". Click on the **OK** button to go back to the Schematic Editor.

10. Select **Option → Export Netlist**. For **Files of type** choose **Edif 200 [*.EDN].** Click on the **Open** button to export the file.

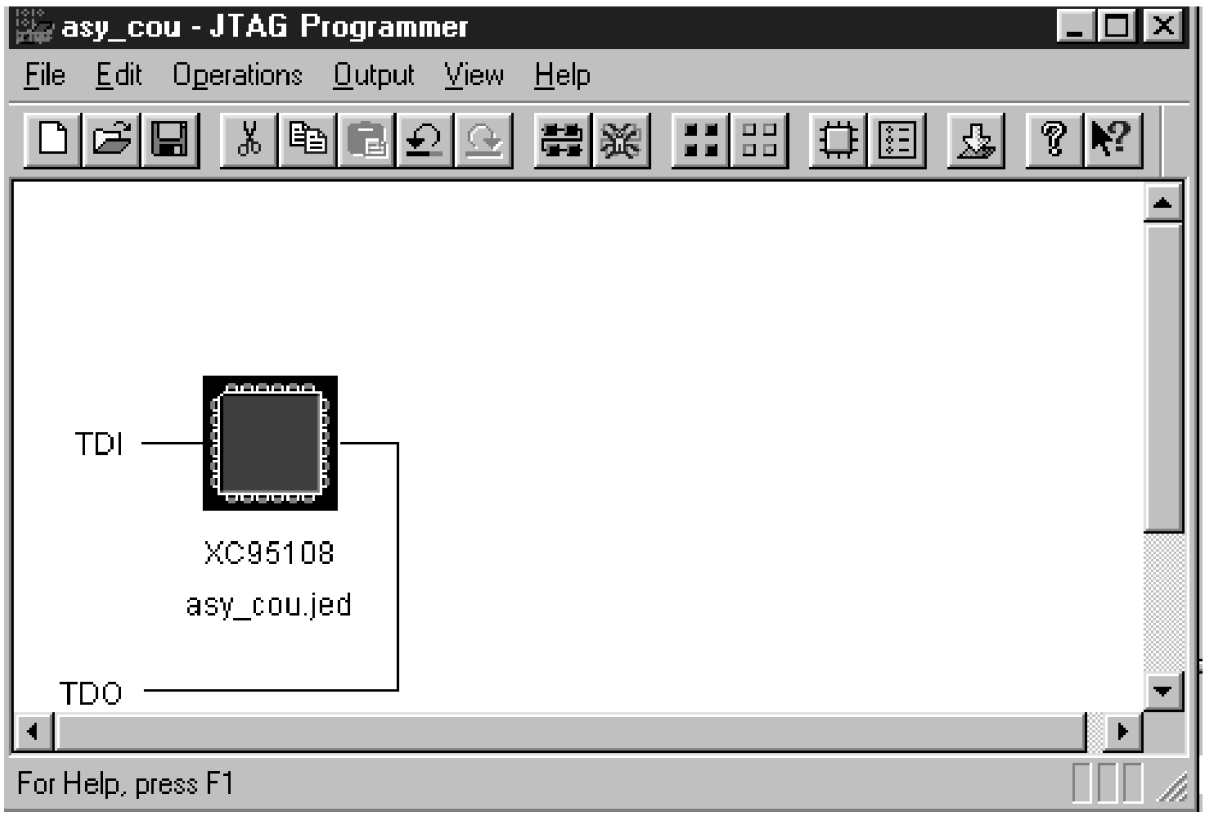
11. The ASY_COU design in the Schematic Editor is now complete. Return to the Project Manager window by selecting **File →Exit**. Verify that a green check mark is on the Design Entry button.

12. Click on the **Implementation** button to compile the design. Verify that the **Device**, **Speed**, **New version name** and **New revision name** edit boxes have ***95108PC84***, ***20***, ***ver1*** and ***rev1***, respectively. Click on the **Run** button.

13. If errors are detected during this stage, the process will be terminated with the message "Implementation completed with errors". Go back to the Schematic Editor and fix them. If there is no error, you will see a window as shown below. Use **File → Close** to go back to the Project Manager and verify that a green check mark appears on the Implementation button.

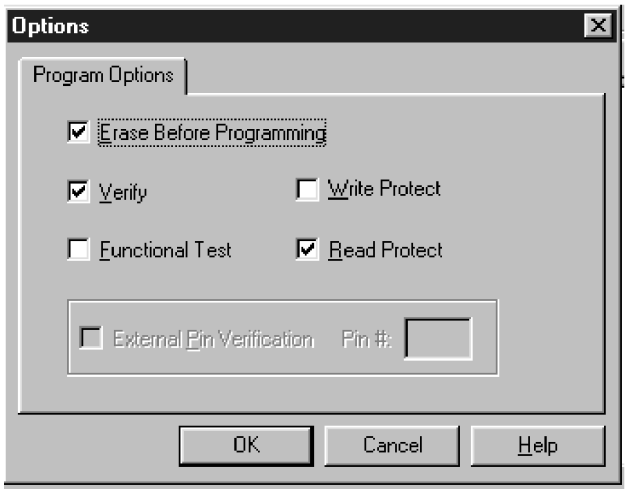
14. Click on **OK** in the message sub-window.

15. Plug the AC adapter into an outlet, and connect it to the PLDT-1 board. Then connect your PC to the PLDT-1 board with the parallel cable provided.




16. To download the program, click on the **Device Programming** button in the Project Manager window, or choose **Tools → Device Programming** on the menu at the top.

17. In the **JTAG Programmer** window, choose **Operations → Program** from the menu bar, the **Options** sub-window should appear. Select **Erase Before Programming** and **Verify**. Then click on the **OK** button.




18. When the download is finished, click on the **OK** button in the **Operation Status** sub-window to go back to the JTAG Programmer window. Then, choose **File → Exit**. At the message sub-window prompt, select **Yes**. In the **Save As** sub-window, click on **Yes** to save the JEDEC file and return to the Project Manager window.

19 Using the jumper wires connect pin #74 on "**HD2**" to "**T2**", Switch "**S1**" will be used as a clock generator. Connect pin #72 on "**HD2**" to "**T4**", switch "**S2**" will be used as a reset switch.

20. Run the program on your PLDT-3 board and fill in the following truth-tables by using the "**S1**" switch for clock and the LEDs for output, for resetting the circuit use switch "**S2**".

**Truth-Table for 3-bit Asynchronous Counter**

| LED 9 | LED 10 | LED 11 | | PULSE SWITCH / S1 |
|---|---|---|---|---|
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |
| | | | | |